A correlation-based entity embedding approach for robust entity linking

Cheikh Brahim El Vaigh, François Torregrossa, Robin Allesiardo, Guillaume Gravier, Pascale Sébillot

A correlation-based entity embedding approach for robust entity linking

Cheikh Brahim El Vaigh
INRIA, IRISA
Rennes, France
cheikh-brahim.el-vaigh@inria.fr

François Torregrossa
Solocal, IRISA
Rennes, France
ftorregrossa@solocal.com

Robin Allesiardo
Solocal
Rennes, France
rallesiardo@solocal.com

Guillaume Gravier
CNRS, IRISA
Rennes, France
guig@irisa.fr

Pascale Sébillot
INSA, IRISA
Rennes, France
pascale.sebillot@irisa.fr

***Abstract*—Entity alignment is a crucial tool in knowledge discovery to reconcile knowledge from different sources. Recent state-of-the-art approaches leverage joint embedding of knowledge graphs (KGs) so that similar entities from different KGs are close in the embedded space. Whatever the joint embedding technique used, a seed set of aligned entities, often provided by (time-consuming) human expertise, is required to learn the joint KG embedding and/or a mapping between KG embeddings. In this context, a key issue is to limit the size and quality requirement for the seed. State-of-the-art methods usually learn the embedding by explicitly minimizing the distance between aligned entities from the seed and uniformly maximizing the distance for entities not in the seed. In contrast, we design a less restrictive optimization criterion that indirectly minimizes the distance between aligned entities in the seed by globally maximizing the dimension-wise correlation among all the embeddings of seed entities. Within an iterative entity alignment system, the correlation-based entity embedding function achieves state-of-the-art results and is shown to significantly increase robustness to the seed's size and accuracy. It ultimately enables fully unsupervised entity alignment using a seed automatically generated with a symbolic alignment method based on entities' names.**

***Index Terms*—Entity Alignment, Dimension Alignment, Knowledge Graphs, Knowledge Graphs Embedding, Information Extraction**

I. Introduction

Knowledge Bases (KBs), often represented as Knowledge Graphs (KGs), provide a powerful resource for reasoning in AI systems and account for domain-specific or encyclopedic knowledge on entities and their relations. The multiplicity of such KBs, not necessarily well-connected in spite of huge efforts as part of the linked open data, however limits their practical use. In this context, entity alignment, a.k.a. entity matching, comes as a practical tool to discover entities in two different KGs that refer to the same real-world concept. Linking KGs is bound to facilitate a variety of tasks such as knowledge extraction and discovery, question-answering, semantic search and knowledge reasoning [12], [13], with practical implications for companies wishing to complete their data warehouse with open data sources.

Conventional methods for entity alignment based on matching symbolic features that describe entities [11], e.g., name, type or attributes, have recently been superseded by learning-based approaches that seek to jointly embed entities from two KGs in a unique space in which the actual linking is performed from the distance between entities of the two KGs [4]–[7].

Most approaches derive from the TransE [1] KG embedding criterion, completed with an alignment objective function. In the context of entity linking between KG_1 and KG_2, joint embedding of entities and relations of the two KGs combines the TransE criterion for semantic embedding with a task-specific criterion, known as calibration. The latter ensures that distance-based linking in the embedding space is meaningful and typically makes use of a *seed* set of known alignments between KG_1 and KG_2 and a *ref* set of entities for which we are searching an alignment.

More specifically, the *seed* alignment provides the correspondence between entities of the two KGs for a few entities and is used to ensure that aligned entities lie close in the embedded space. The seed remains today the major bottleneck of state-of-the-art techniques: its size and quality have a substantial impact on the alignment accuracy, requiring human expertise in practice to create accurate seeds. In contrast, the *ref* entities are a set of entities from the two KGs for which we search a pairing. In academic work, the alignment on the *ref* set is known and used for evaluation purposes. Embedding techniques for entity alignment indirectly make assumptions on the *ref* alignments. For instance, the iterative approach [7] assumes a uniform distribution of entity distances in the *ref*; similarly, [10] performs negative sampling only for entities in the *ref*. Clearly, the hypothesis claiming that distance between pairs of entities not in the seed should be uniformly scattered is wrong for truly aligned entities, and thus potentially harmful.

In this work, we propose and evaluate a novel calibration criterion for entity alignment that, combined with the TransE semantic criterion, yields a robust and efficient entity alignment system. Contrary to previous work, our calibration criterion does not directly seek to minimize the distance between aligned entities in the seed but rather addresses the task indirectly. We globally seek to maximize the correlation between embeddings of seed entities in KG_1 and their corresponding counterpart in KG_2 across dimensions of the embedding space. This indirectly lowers the distance between aligned entities and regulates the embedding's dimensions: different dimensions must be uncorrelated. The interest of this indirect approach is mostly twofold: (a) we make no assumptions whatsoever on the set of *ref* entity alignments, thus alleviating the issue of wrong assumptions made by other methods; (b) errors in the seed alignments only have limited impact (if in reasonable

amount obviously) because of the indirect manner in which we approach the task. This new calibration strategy ultimately leads to a new methodology that replaces hand-crafted seed generation with automatic seed generation relying on symbolic methods, with the aim to exclude human input as much as possible from the alignment process.

The main contributions of the paper can be summarized as follows:

1) *a novel iterative embedding learning technique for entity alignment*, named AlignD, leveraging dimension alignment within the joint embedding space;
2) *detailed experiments showing the effectiveness of AlignD* w.r.t. other state-of-the-art methods;
3) *an unsupervised approach to efficiently align entities from scratch*, i.e., without human-generated seed, reaching performance similar to supervised state-of-the-art methods with no human input at all.

Our source code, datasets and experimental results are made available online for reproducibility purposes[1].

The paper is organized as follows: Sec. II discusses the state of the art, introducing concepts and notations required for the description of our method in Sec. III. Experimental results are grouped in Sec. IV before concluding remarks and future work in Sec. V.

II. Related Work and Notations

This section introduces an overview of the prominent methods from the literature. In the following, the *seed* (resp. *ref*) refers to a given set of aligned entities, i.e., a correspondence between an entity of KG_1 and one of KG_2.

A. Overview of Existing Work

Historically, automated entity alignment initially leveraged various symbolic features of KGs such as their properties and entities attributes. Those approaches face the issue of heterogeneity between KGs, in particular different languages and schemas. To skirt the issue, a few approaches also make use of external lexicons, machine translation, and Wikipedia links [11] to help match properties and attributes across KGs, yet remaining difficult to generalize and scale.

The past few years have seen the fast emergence of embedding-based approaches based on representation learning and exhibiting better performance and generalization capabilities than symbolic approaches. All such methods combine knowledge graph embedding with an entity alignment objective function, leveraging two broad families of knowledge graph embedding techniques, namely TransE [1] and graph convolution networks (GCNs).

In the first family, MtransE [6] learns the embedding of each KG independently using TransE [1], and proposes different transformations to perform the alignment between the two embeddings. IPtransE [4] iteratively learns a joint embedding of KGs, and integrates three modules (translation-based, linear transformation and parameter sharing) for entity alignment. JAPE [5] further improves entity alignment by introducing attribute correlations in the process of KG embeddings. BootEA [7] interestingly proposes a constrained version of TransE, adding an explicit criterion to minimize the distance between aligned entities from the seed. BootEA also iteratively uses the links inferred between the KGs to progressively improve the alignment in a semi-supervised learning manner.

The second family leverages graph convolution networks (GCNs) instead of the TransE objective [8]–[10]. A straightforward, yet efficient, application of graph embedding is used in [9] to jointly embed the entities to align. MuGCN [8] additionally performs graph completion similar to KG saturation. Finally, RDGCN [10] builds a dual relation graph of the original KGs put together, a procedure similar to the notion of parameter swapping (see Sec. II-B for details on parameter swapping) and makes them interacting before jointly embedding entities through the dual graph. The construction of the latter renders the approach very sensitive to the quality of the seed, aligned entities in the seed affecting the dual graph topology.

Globally, the main drawback of embedding-based techniques is the need for a high-quality seed, which is used to ensure the quality of the joint embedding space and, ultimately, maximize the accuracy of the final alignment on *ref*. As all methods directly and explicitly rely on the alignment provided in the seed, they are sensitive to the size and quality of the seed.

B. Background Notations and Technical Details

We now provide further technical details on state-of-the-art embedding-based alignment, which we will make use of in the experimental section, introducing notation and highlighting the limits that we address.

1) Entity embedding objective function: Formally, the KG embedding objective function, which takes care of the semantics of the KGs, is defined in the case of TransE [1] as

$$O_e = \sum_{\tau \in T^+} [f(\tau) - \lambda_1]_+ + \mu_1 \sum_{\tau' \in T^-} [\lambda_2 - f(\tau')]_+ \ , \quad (1)$$

where $f(\cdot)$ is a triple scoring function, here $f((h,r,t)) = ||v_h + v_r - v_t||_2^2$, μ_1, λ_1 and λ_2 are hyper-parameters, and $[x]_+ = max(x, 0)$. T^+ and T^- denote the sets of positive and negative triples respectively. The set of the latter, T^-, is obtained by replacing the head or the tail of an existing triple with another non-related entity. In practice, *parameter swapping* is used as a standard practice to enforce similar properties between the two KGs, augmenting the set of positive triples T^+ by injecting triples of KG_1 into KG_2 and vice-versa based on the seed alignment [4], [5], [7].

2) Alignment objective function: For alignment purposes, the objective function O_e is combined with an objective function O_a that measures the discrepancy between the vectors of the aligned entities. The actual form of the alignment objective function depends on the method [2], [4]–[7]. Yet, all make direct use of the seed alignment and indirect assumptions on the alignment of the *ref* entities. For entities in the seed,

[1] https://gitlab.inria.fr/celvaigh/alignd

the distance between the respective embeddings of the entities is minimized. For entities in the *ref*, the objective function typically seeks to uniformly maximize the distance between any pair of entities, leveraging negative sampling in [10] or a likelihood matrix in [7]. This general scheme emphasizes two major limitations that we address in this paper: (a) the distance between seed entity pairs is minimized batch-wise or individually, which prevents from modelling global alignment between embeddings and makes the approach sensitive to errors in the seed; (b) the hypotheses on other entities, which mix entities that appear in the two KGs and should thus be aligned with entities that have no counterpart in the other KG, are too strong, considering the two types of entities on equal foot.

3) Iterative alignment: Iterative alignment, a.k.a. *alignment bootstrapping*, casts the problem of entity alignment in a semi-supervised learning approach that alternates embedding learning and alignment inference from the embedding. For each training iteration, given a current embedding obtained from the combination of the two objective functions discussed above, the idea is to choose the most confident matching pairs to predict an alignment of entities in *ref*. A one-to-one mapping constraint is added in [7] and used to improve the alignment prediction. This predicted alignment is then used in addition to the seed alignment in the re-estimation of the embedding, impacting parameter swapping and the optimization of the combined objective function.

III. AlignD

To address the limits of the alignment objective functions that we presented in the previous section, we propose a novel alignment criterion that globally considers seed entities rather than iterating through the pairs of aligned entities, and that do not make assumptions on the *ref* entity pairing. The general idea of our approach is inspired by [3] and relies on Pearson's correlation coefficient between the dimensions of the embedding of aligned entities, with the idea of globally maximizing the correlation between corresponding dimensions as an indirect way to move the seed's aligned entities closer in the embedded space. Our aim is to compel dimension alignment of both embeddings such that aligned entity pairs are easily identifiable through cosine similarity by enforcing embedding dimensions to represent identical latent information and disentangle different dimensions.

A. Measuring Embedding Alignment

At the core of our approach is a measurement of how well the embeddings—E_1 and E_2 both with dimension D—of KG_1 and KG_2 match by looking at the correlation between dimensions of the embeddings over the matching entities in the seed. To this end, we define the correlation between two dimensions in the embedded space as Pearson's correlation between the coordinate of entities on the first dimension and the coordinate of the aligned entities on the second dimension. In other words, we quantify the dependency between the value of an entity embedding on the i-th dimension and the value of the corresponding (aligned) entity on the j-th dimension. The seminal idea, deriving from the fact that we use cosine similarity between embeddings to perform entity alignment, is that corresponding dimensions in E_1 and E_2 should match and carry the same information, which we designate as dimension alignment.

As we are interested in dimension alignment between the embeddings of the two different KGs, we rely on the correlation matrix where rows are indexed by the dimensions of the first embedding and columns by the dimensions of the second one. Entities are identified according to a seed set S of size N, such that the correlation coefficient at coordinates i, j models the interaction between the values of N entities of KG_1 on dimension i and the values of the corresponding N entities of KG_2 on dimension j. The idea is that dimensions i from KG_1 and j from KG_2 are likely to be aligned if the values of entities on those dimensions are significantly proportional. Formally, we write the correlation matrix A_D as

$$A_D = (r(E_1(S)_i, E_2(S)_j))_{(i,j)\in[\![1,D]\!]^2} \ ,$$

where $r(E_1(S)_i, E_2(S)_j)$ denotes the Pearson's correlation coefficient between dimension i of E_1 and j of E_2 on the entities from the seed set S.

Following this line of thought, a global indicator of dimension alignment quality is derived from A_D by observing two simple facts: on the one hand, E_1 and E_2 are aligned if dimension i in the embedding E_1 corresponds to dimension i in the embedding E_2, i.e., they are positively correlated; on the other hand, correlations on identical dimensions should be preponderant with respect to correlations between distinct dimensions. In other words, non diagonal terms in A_D should be small with respect to diagonal terms and hence the distance between A_D and I, the identity matrix, provides an approximation of the dimension alignment between the embeddings of the two KGs. In this work, we define the following criterion based on the L2-norm as a measure of dimension alignment discrepancy

$$O_d = \|A_D - I\|_2 \ . \tag{2}$$

High values of O_d means large deviation from the situation above and hence poor correlation between corresponding dimensions in the two embeddings, dimension alignment being obtained by minimizing O_d at learning.

O_d looks at vector coordinates for every couple in S simultaneously on each individual dimension through Pearson's correlation, thus forcing the representations to be globally consistent. The aim of O_d is thus twofold: (a) it ensures that each dimension of the embeddings E_1 and E_2 is consistent and avoids two distinct dimensions to be proportional, encouraging the encoding of distinct information in distinct dimensions; (b) it reduces the sensitivity to noisy couples by processing all couples in S at once, erroneous examples (if in minority) thus tend to be ignored.

B. AlignD Objective Function and Algorithm

Based on the measure of dimension alignment defined by Eq. 2, we propose a novel algorithm based on an iterative

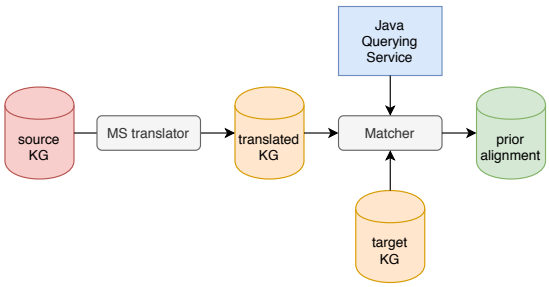

Fig. 1: Pipeline used to automatically compose a seed.

alignment philosophy of [7]. In the AlignD entity alignment system, the joint embedding of the two KGs is obtained by minimizing the combined objective function

$$O = O_e + \alpha O_d \ , \tag{3}$$

where O_e is the TransE entity embedding objective function as defined in Eq. 1, O_d is the correlation-based alignment objective function as defined in Eq. 2 and α is a balance hyper-parameter. The combination of this new objective function O from Eq. 3 with parameter swapping and iterative alignment, as described in Sec. II-B, leads to our alignment procedure AlignD. Both bootstrapping and the final alignment are computed leveraging the cosine similarity between embeddings. It is worth noting that the objective function O makes use of all entities in the embedding objective O_e yet not making specific assumptions for entities not present in the seed as O_d is obtained only from the seed alignments.

It is known that the size of the seed substantially influences the alignment accuracy, in particular, increasing the seed's size is likely to improve alignment of *ref* entities. This is particularly true for AlignD which do not put constraint on entities not in the seed, contrary to other methods. Hence, increasing the seed size as no side effects on the remaining entities and can only be beneficial. This suggests that the seed can be extended with automatic alignment methods, yielding a larger seed however with a small amount of incorrectly aligned entities. Because of the global vision of all entities in the seed (no batching) through O_d and of the absence of hypotheses on entities not in the seed, AlignD is designed to make the most of a large but not fully accurate seed, as long as the fraction of incorrect alignments in the seed remains reasonable. On the contrary, other embedding-based alignment methods, such as BootEA or RDGCN, would be greatly affected by an extended or noisy seed since they enforce aligned entity vectors to perfectly match and assume uniformity of similarities for entities not in the source seed.

C. Automatic Seed Replacement

Matching KGs using embedding methods fails to provide an accurate alignment without a seed. Pushing the idea of automatically extending the seed alignments to an extreme case, we propose to replace the expert-based generation of a seed with a fully-automatic approach where the seed is generated automatically. In particular, we study a set of symbolic alignment approaches, which do not need manually-generated seeds, and use their predictions as a starting point in AlignD. Those symbolic matching approaches rely on basic string comparison between entities' names. The whole idea is to select the couples of entities from the KGs that have strong similar labels according to some string metric. The process to automatically align entities from their labels, and thus predict a seed, is shown in Fig. 1, where we fix KG_1 to be the source KG and KG_2 the target one. Since the KGs are not necessarily in the same language, a translator module is used on the source KG, namely Microsoft Translator in our experiments. For every entity from the source KG, a list of entities from the target KG with a similar name is retrieved and the one with the highest string similarity is selected, provided the similarity is higher than a threshold to keep only relevant matches. Beside the standard string matching, we tried fuzzy matching to increase the recall of predicted aligned entities. We also studied four other string metrics, namely the Sørensen–Dice coefficient (DSC), similar to a Jaccard index, the Levenshtein distance, the Jaro–Winkler distance that gives more favorable ratings to strings matching from the beginning, and the cosine similarity using a pre-trained word embedding. This methodology, combining embedding-based entity alignment and symbolic approaches, allows KGs alignment without prior knowledge.

IV. Experiments

Experimental validation is conducted on standard benchmarks, comparing AlignD with the state of the art and showcasing its robustness with respect to the seed. The standard dataset DBP15K [5] have been used in our experiments. It contains three cross-lingual datasets built from the multilingual versions of DBpedia: Chinese to English (ZH-EN), Japanese to English (JA-EN) and French to English (FR-EN). Each dataset contains 15,000 aligned entities.

Following previous work on entity alignment, e.g., [5], [7], KG embeddings are trained with a seed containing 30 % of the alignments, the remaining ones being used for test purposes (*ref*). For evaluation, we classically report the mean reciprocal rank (MRR) and Hits@k after ranking all entities of KG_2 for each given entity of KG_1 according to the cosine similarity: Hits@1 strongly correlates with the quality of the alignment, while Hits@10 and MRR mostly indicate the quality of the embedded space for the alignment process.

Two variants of AlignD were implemented:

1) AlignD or AlignD(*glove*,300), where a perfect seed is used for the training, allowing to compare AlignD with state-of-the art approaches. See details on AlignD(*glove*, 300) in Sec. IV-A.
2) AlignD[X] where AlignD is trained with a seed extended or automatically generated by an algorithm X.

For all experiments, the following hyper-parameters were chosen for KG embeddings with AlignD: $\lambda_1 = 0.01$, $\lambda_2 = 2$ and $\mu_1 = 0.2$. The parameter α depends on the seed and is set to the size of the seed at hand. The learning rate was set to 0.01, the training to 500 epochs, with semi-supervised alignment

bootstrapping every 10 epochs. These hyper-parameters are chosen according to previous experiment settings reported in the literature, notably BootEA [7]. Finally the embedding dimension was set to 75, except for AlignD($glove, 300$) and AlignD[RDGCN,seed] where the embedding dimension is 300. This is due to the pre-trained embeddings used in RDGCN [10], for which very few information is provided.

A. Comparison to State-of-the-Art Approaches Using a Ground-Truth Seed

We first compare our approach to a series of entity alignment systems which reported state-of-the-art results in recent years on the DBP15K datasets, namely:

- MTransE [6] which learns a transformation between two fixed embeddings;
- IPtransE [4] which iteratively learns joint embeddings of KGs using PTransE [2];
- JAPE [5] which learns the embedding of KGs jointly while preserving entities attributes;
- GCN-based approaches MuGCN and RDGCN [8], [10], where the former uses graph completion to improve the matching while the latter builds a dual of KG_1 and KG_2 unified. RDGCN is our GCN baseline;
- BootEA [7], a semi-supervised technique which iteratively labels KG_1 entities with KG_2 entities, and which we consider as our iterative baseline.

All results are gathered in Tab. I, using for all methods the same error-free seed corresponding to 30 % of the existing alignments. Results in rows 1 to 6, with embedding dimension 75, clearly show AlignD to be comparable or slightly better than the BootEA baseline on all datasets. RDGCN is not directly comparable to the other approaches in Tab. I, as it uses pre-trained word embeddings of dimension 300, used to initialize embeddings of entities based on their name. To allow fair comparison, we thus trained AlignD using the same initialization as RDGCN, denoted AlignD(*glove*,300). In comparable conditions, AlignD performs significantly better than RDGCN.

B. Automatically Extended Seed

We now study the impact of training AlignD with a seed predicted using an off-the-shelf embedding-based entity alignment approach. Therefore, AlignD was trained using the prediction of BootEA and RDGCN, both obtained from a ground-truth seed (containing 30 % of the alignments) at the initial iteration of these last two algorithms. The initial alignment as input to AlignD can thus be seen as the initial seed extended with BootEA or RDGCN, hence the notion of extended seed.

Results on the DBP15k datasets are given in Tab II. We can see that AlignD[BootEA] (i.e., AlignD with initialization obtained from BootEA) performs better than AlignD and BootEA alone, due to the seed extension. We also compared AlignD[RDGCN] with RDCGN, both using GloVe pre-trained embedding for entity names embedding as initialization. As previously, applying AlignD following RDGCN improves over the latter, however not being significantly better that AlignD(*glove*, 300) directly applied on the seed. We believe this is due to the high-quality of the GloVe initialization.

Globally, every extended seed experiment improves significantly the performance w.r.t. their corresponding experiment with a ground-truth seed, the best results over all our experiments being obtained with AlignD[RDGCN]. We thus conclude that using the prediction of a matching algorithm to artificially extend the size of seed leads to a more accurate final alignment with AlignD, exploiting its robustness to noise.

C. Automatically Generated Seed

Increased robustness of AlignD to errors makes it possible to overcome the need for prior alignment between KGs, replacing human-based seed alignments generation by substring matching between entities' names. Our objective here is to demonstrate the capacity of AlignD in designing a system to align KGs with no ground-truth seed at all, combining symbolic methods with embedding-based ones.

Tab. III, symbolic columns, gathers results for the six symbolic methods mentioned in Sec. III-C on the DBP15k datasets. The best symbolic approaches are substring matching for FR-EN, Jaro-Winkler for JA-EN and fuzzy matching for ZH-EN.

Symbolic matching of entity names is here used to provide an initial, low quality, seed for AlignD. We first compare in Tab. III, last three columns, the impact of the seed generation approach on AlignD. AlignD not only improves the results of all the symbolic methods but reduces the discrepancies between them. The best Hits@1 score is obtained when combining Jaro-Winkler with AlignD. This is due to the fact that Jaro-Winkler gives more favorable rating to strings that match from the beginning. Entity names in DBpedia, which are unique and coherent from one language to another, are thus well-matched with this method.

We further compare in Tab. IV the best unsupervised system combining a symbolic approach with AlignD (AlignD[JaroWinkler]) with the state-of-the-art entity alignment methods applied on the same substring matching seed alignment. AlignD[JaroWinkler] outperformed all the methods, taking advantage of its robustness to errors in the seed, except AlignD($glove, 300$)[JaroWinkler]. The latter obtained the best results, outperforming the state of the art. Interestingly, AlignD without pre-trained word embeddings initialization, got close score to the unsupervised methods using pre-trained word embeddings, which means that GloVe initialization only adds limited gain when using automatically generated seed while being hard to obtain in real world scenario. These good results are partly explained by the size of the initial alignment provided by the string matching method, which is much larger than the 30 % of the alignments used by other methods, however much noisier. We noticed the complementary performance of the two methods as AlignD($glove, 300$)[JaroWinkler] is better on ZH-EN and FR-EN while AlignD[JaroWinkler] is better on JA-EN, even if they do not have either the same dimension size or the same initialization. As AlignD[JaroWinkler]

TABLE I: Hits@1, Hits@10 and MRR with ground-truth seed on DBP15K datasets. The top half results are taken from the literature while we produced bottom ones by running new experiments.

Approaches	ZH-EN			JA-EN			FR-EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE [6]	30.83	61.41	0.364	27.86	57.45	0.349	24.41	55.55	0.335
IPTransE [4]	40.59	73.47	0.516	36.69	69.26	0.474	33.30	68.54	0.451
JAPE [5]	41.18	74.46	0.490	36.25	68.50	0.476	32.39	66.68	0.430
MuGCN [8]	49.56	87.03	0.621	50.10	85.70	0.621	49.50	87.00	0.621
BootEA	61.89	84.01	0.695	57.43	82.93	0.661	58.31	84.83	0.676
AlignD	62.68	84.70	0.701	58.88	83.06	0.671	60.77	85.30	0.691
RDGCN	70.75	84.55	–	76.74	89.54	–	88.64	95.72	-
AlignD($glove, 300$)	**82.30**	**93.93**	**0.864**	**84.90**	**94.24**	**0.882**	**90.85**	**97.26**	**0.913**

TABLE II: Hits@1, Hits@10 and MRR with an automatically extended seed.

Approaches	ZH-EN			JA-EN			FR-EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
AlignD[BootEA]	64.66	85.10	0.715	62.82	85.60	0.704	65.94	87.15	0.732
AlignD[RDGCN]	**82.20**	**93.97**	**0.863**	**83.84**	**94.07**	**0.874**	**90.38**	**97.03**	**0.926**

TABLE III: Hits@1 using symbolic methods alone or in combination with AlignD.

Approaches	symbolic			AlignD[symbolic]		
	ZH-EN	JA-EN	FR-EN	ZH-EN	JA-EN	FR-EN
DSC	25.22	44.64	54.10	80.28	84.01	89.33
Fuzzy	**35.00**	44.93	46.57	71.05	81.81	84.97
SubString	24.75	44.29	**57.00**	70.79	76.84	85.95
Word2Vec	24.38	37.49	39.86	79.61	80.81	84.47
Levenshtein	25.07	44.27	54.05	79.95	84.08	89.2
JaroWinkler	29.06	**49.01**	54.73	**81.21**	**85.30**	**89.61**

TABLE IV: Hits@1 for alignment methods with seed automatically generated with a symbolic approach.

Approaches	ZH-EN	JA-EN	FR-EN
BootEA[JaroWinkler]	61.35	57.56	59.06
RDGCN[JaroWinkler]	65.10	75.76	83.52
AlignD[JaroWinkler]	81.21	85.30	89.61
AlignD($glove, 300$)[JaroWinkler]	**83.12**	**84.58**	**92.15**

requires neither human intervention for seed generation nor initialization, we recommend its use rather than that of AlignD($glove, 300$)[JaroWinkler] in real-world contexts.

V. Conclusion and Future Work

Approaches in KG alignment from embeddings rely on a prior set of aligned entities shared by the KGs, often manually provided by experts. By introducing the alignment of the dimensions of the initial KG embedding spaces in the learning process, as an indirect criterion to embed similar entities together, we showed that we can limit the need for prior high-accuracy alignment paving the way towards a fully-automatic knowledge-free entity alignment system. Globally, our proposition improves the effectiveness with regard to state-of-the-art methods on inaccurate seeds and answers a wider range of realistic scenarios where perfect handcrafted seeds are not available.

This contribution opens up new horizons towards fully exploiting the semantics of RDF knowledge bases for entity alignment. In particular, the dimension alignment criterion that we propose can be combined with RDF embeddings other than TransE, including graph-based ones. For future work, instead of string similarities, it will be useful to use other entity attributes to automatically build prior alignment, in order to extend and improve the quality of the seed. We plan also to examine other more robust differentiable correlation coefficients in future works instead of Pearson's correlation coefficient.

References

[1] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," in NIPS, 2013, pp. 2787–2795.

[2] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao and S. Liu, "Modeling Relation Paths for Representation Learning of Knowledge Bases," in EMNLP, 2015, pp. 705–714.

[3] Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample and C. Dyer, "Evaluation of Word Vector Representations by Subspace Alignment," in EMNLP, 2015, pp. 2049–2054.

[4] H. Zhu, R. Xie, Z. Liu and M. Sun, "Iterative Entity Alignment via Joint Knowledge Embeddings," in IJCAI, 2017, pp. 4258–4264.

[5] Z. Sun, W. Hu and C. Li, "Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding," in ISWC, 2017, pp. 628–644.

[6] M. Chen, Y. Tian, M. Yang and C. Zaniolo, "Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment," in IJCAI, 2017, pp. 1511–1517.

[7] Z. Sun, W. Hu, Q. Zhang and Y. Qu, "Bootstrapping Entity Alignment with Knowledge Graph Embedding," in IJCAI, 2018, pp. 4396–4402.

[8] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li and T.-S. Chua, "Multi-channel Graph Neural Network for Entity Alignment," in ACL, 2019, pp. 1452–1461.

[9] Z. Wang, Q. Lv, X. Lan and Y. Zhang, "Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks," in EMNLP, 2018, pp. 349–357.

[10] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan and D. Zhao, "Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs," in IJCAI, 2019, pp. 5278–5284.

[11] P. Shvaiko and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," IEEE TKDE, 2013, vol. 25, no. 1, pp. 158–176.

[12] E. Brill, J. Lin, M. Banko, S. Dumais and A. Ng, "Data-Intensive Question Answering," in TREC, 2001, pp. 393–400.

[13] R. Guha, R. McCool and E. Miller, "Semantic Search," in WWW, 2003, pp. 700–709.